情報処理学会論文誌 Vol.65 No.1 83–96 (Jan. 2024)

セマンティックセグメンテーションを利用した GAN Inversionによる背景画像の編集手法の提案

石幡 柊介[1,a)] 折原 良平[1,b)] 清 雄一[1,c)] 田原 康之[1,d)] 大須賀 昭彦[1,e)]

受付日 2023年3月14日, 採録日 2023年10月3日

概要：近年，StyleGAN を画像編集タスクに適用する研究が進められている．画像編集タスクは背景画像の編集にも適用可能だが，背景画像は顔画像などの前景画像に比べて多様であるため，画像の編集性能が低下する．また，編集内容を的確にシステムに伝えることが難しいため，コンテント編集が困難という問題もある．たとえば自然言語による画像編集では編集対象となる背景画像のオブジェクトの指定が曖昧となるため，編集された画像は編集者にとって好ましくないものとなってしまう．一方でセマンティックセグメンテーションを使用すれば編集者の意図するコンテントの編集ができると考える．本研究では GAN Inversion と呼ばれるタスクにおいて，セマンティックセグメンテーションマスクを取り入れた，エンコーダベースの GAN Inversion 手法である HyperStyle を基にしたフレームワークを提案する．GAN Inversion で求められる画像の再構成品質を維持しつつ，従来のスタイル編集性能を持ちながら，コンテント編集も可能にする．実験を行った結果，定性的な評価では本モデルが画像のコンテントとスタイルを別々に編集できることを確認した．

キーワード：StyleGAN，GAN Inversion，画像編集

Background Image Editing Method by GAN Inversion with Semantic Segmentation

Syuusuke Ishihata[1,a)] Ryohei Orihara[1,b)] Yuichi Sei[1,c)]
Yasuyuki Tahara[1,d)] Akihiko Ohsuga[1,e)]

Received: March 14, 2023, Accepted: October 3, 2023

Abstract: Recently, research has been conducted on applying StyleGAN to image editing tasks. Although the technique can be applied to editing background images, because they are more diverse than foreground images such as face images, editability is compromised. In addition, content editing is difficult because it is difficult to accurately convey the edited content to the system. For example, because natural language instructions can be ambiguous, edited images become undesirable for the user. Therefore, a semantic segmentation mask can be used to edit content as intended by the editor. In our study, we propose a framework based on HyperStyle, an encoder-based GAN Inversion method that incorporates a semantic segmentation mask in a task called GAN Inversion. Our method can edit the image style and content independently while maintaining the quality of image reconstruction required by GAN Inversion. As a result, the qualitative evaluation confirms that our model enabled the editing of image content and style separately.

Keywords: StyleGAN, GAN Inversion, image editing

1. はじめに

近年，深層学習を用いた画像の自動生成では Generative

[1] 電気通信大学大学院情報理工学研究科情報学専攻
University of Electro-Communications Graduate School of Informatics and Engineering Department of Informatics, Chofu, Tokyo 182–8585, Japan
[a)] ishihata.syuusuke@ohsuga.lab.uec.ac.jp
[b)] orihara@acm.org
[c)] seiuny@uec.ac.jp
[d)] tahara@uec.ac.jp
[e)] ohsuga@uec.ac.jp

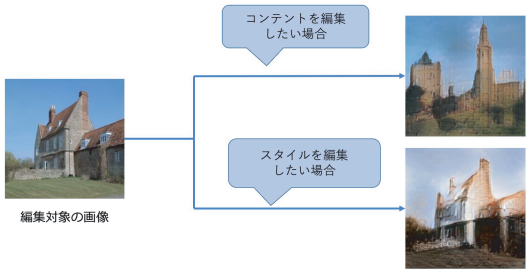


図 1 画像編集に関する研究の概要．画像に含まれるコンテント，スタイルのいずれか，または両方を編集することを目的とする

Fig. 1 Overview of our research on image editing. We aim to edit either the content, style or both in the image.

Adversarial Networks（GAN）[2] をはじめとして，大きな進歩をとげている．その中でも，教師なし学習の GAN モデルの 1 つである StyleGAN [15] は，髪の毛や目の色といった属性どうしを分離し，高品質な画像の生成ができる．この StyleGAN は画像間の補間性能に優れているため，その表現力の高さを画像編集タスクに利用する研究事例が増えている．たとえば StyleCLIP [17] は，自然言語テキストを入力して学習済み StyleGAN の潜在変数をテキストの内容に即した画像に編集する研究である．CLIP [24] という自然言語と画像の分類に利用されるモデルを損失関数に用いることによって潜在変数の編集を行う．また，GAN Inversion は，GAN の Generator が入力画像を再構成するような潜在変数を推定するタスクである．推定された潜在変数を操作することで対象の画像を編集することができる．本研究では編集対象の画像をソース画像，編集に使用する情報を持つ画像を参照画像と呼ぶ．

画像はコンテントとスタイルという情報を持っており，コンテントは画像内の物体の形状や構造，スタイルは絵柄や画風を表している．コンテント編集は画像のスタイルを維持したままコンテントのみを編集することであり，スタイル編集は画像のコンテントを維持したままスタイルのみを編集することである．ソース画像はコンテント編集のときは参照画像のコンテント，スタイル編集のときは参照画像のスタイルをもとに編集される．

カメラから被写体までの絶対的な距離が遠い写真を背景画像とし，それを本研究の対象とする．一方，カメラから被写体までの距離が近い画像を前景画像と呼ぶ．前景画像の例は顔画像である．顔画像を編集する際には，目や鼻などのパーツの位置がある程度固定されていなければならない．このように前景画像については編集に制約が存在する．それに対して，背景画像ではそのような制約は少ないため，多様性のある柔軟な編集が求められる．背景画像では，**図 1** のようにコンテントとスタイルを別々に編集することが有用である．たとえば，1 枚の写真を編集して様々な画像を生成できるため，写真集や映像作品の制作時間を短縮することができる．しかし，背景画像は前景画像に比

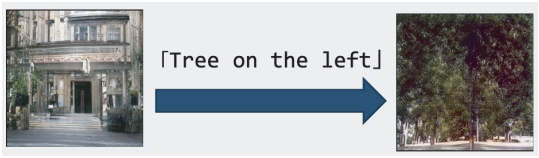


図 2 StyleCLIP で背景画像を編集した例．テキストプロンプトは ‘Tree on the left’

Fig. 2 Example of editing a background image in StyleCLIP. The text prompt is ‘Tree on the left’.

べて多様であるため，GAN の生成する画像の品質が低下する．それにともない，編集性能も低下してしまう．

また，コンテントの編集が難しいという問題もある．たとえば，StyleCLIP で画像のスタイルを編集することは可能であるが，画像のコンテントを指定する場合，直感的に編集することが困難である．その結果，オブジェクトの位置のズレや前後関係の違いといった編集者の意図とは異なる効果として画像に現れてしまうことがある．そこで，StyleCLIP でコンテントを考慮した画像編集が可能かどうかを確認した．**図 2** では，本来なら木が左側にくるように編集されるはずの入力画像が，画像全体が木に覆われた画像に編集されており，テキストプロンプトの “left” の部分が無視されている．コンテントを考慮した編集が不十分であった．

GAN Inversion タスクでは，生成画像の再構成品質と編集性能を両立させることが課題である．そのために，Generator のパラメータを変更する HyperNetworks を用いたアプローチがある．再構成品質は Generator で生成した画像がどれだけ入力画像を再現しているかを示している．この品質が低いと，入力画像を再現する潜在変数が推定できていないため，編集者が所望する画像編集は難しい．背景画像の場合，pSp [19] のようなエンコーダベースのアプローチでは，再構成品質が低くなる．また，背景画像の多様性により，Generator の性能が低下する．HyperNetworks [13] は Generator の性能を向上させるため，この問題を解決することができる．HyperStyle [10] と呼ばれる GAN Inversion のアプローチは，HyperNetworks の出力である残差パラメータを用いて Generator の重みを調整し，再構成品質を向上させるものである．背景画像では，残差パラメータによる Generator の調整の影響により，画像全体の形状が変化することが分かった．

テキストではできなかったコンテント編集に対して，セマンティックセグメンテーションマスクは編集者が意図したコンテントを視覚的に表現するため，コンテントの編集に便利であると考えた．そこで，StyleGAN のスタイル編集性能を維持しつつ，セマンティックセグメンテーションマスクを用いてコンテントの編集をする HyperStyle ベースのフレームワークを提案する．GAN Inversion タスクでは，再構成品質と編集性能の 2 つの軸で性能を評価する．

どちらの軸も重要であるが，本研究では特に編集性能に着目しており，特に画像のコンテントとスタイルを独立して編集できるか否かに重点をおいている．画像編集実験では，提案モデルが画像のコンテントとスタイルを別々に編集できることを定性的に確認した．

また本論文では，2 章では，関連研究，3 章では提案手法の説明，4 章では提案手法を用いた実験と評価について，5 章では考察，最後に 6 章で本論文の結論をまとめ，今後の展望を示す．

この論文は国際会議 ICAART で発表した内容を拡張したものである[*1][1].

2. 関連研究

2.1 GAN

Generative Adversarial Networks（GAN）とは深層学習の手法の 1 つである．これは生成器（Generator）と判別器（Deltaiscriminator）の 2 つのニューラルネットワークを用いている（**図 3**）．Generator にはランダムベクトル z を入力に渡し，学習データに基づいたデータを生成する．そして Discriminator にその生成データと学習データを入力して生成データが学習データなのかどうかの判別をする．この Generator と Discriminator を互いに学習させ，Generator は Discriminator に対して生成データを学習データと認識させるように学習して，生成データの精度を向上させる．

このアプローチに基づき，GAN の性能を向上させるために Convolutional Neural Network（CNN）を用いた DCGAN [3] をはじめとして様々な画像生成，変換アプローチが提案されている．画像変換の例として，白黒画像や線画をカラー画像に変換する手法のほかに，セマンティックセグメンテーションマスクから合成するアプローチも存在する．SPADE [7] や SEAN [26] は正規化を工夫したアプローチで，それはセマンティックセグメンテーションを取り入れてそのセグメンテーションの形状に応じて画像を合成するものである．これらの手法はセグメンテーションによって画像のコンテントを編集者の意図するコンテントに編集することができるが，スタイルを維持することは困難である．本研究ではスタイルを維持したまま編集者の意図するコンテントに編集することが可能である．

GAN の Generator，特に StyleGAN では，入力に用いる潜在変数によって生成する画像が決まるため，潜在変数を操作することで画像を編集することが可能である．さらに StyleCLIP [17] のようなテキストによる画像編集のアプローチがある．

[*1] 情報処理学会のポリシとして国際会議論文は途中経過報告と見なされるため，二重投稿にならないことを確認している．

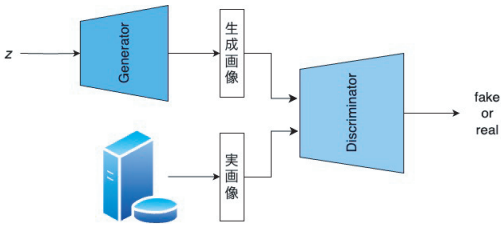


図 3 GAN のアーキテクチャ
Fig. 3 Overview of GAN architecture.

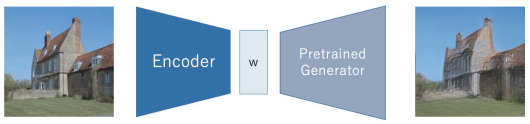


図 4 一般的なエンコーダベースの GAN Inversion の概要．エンコーダは事前に学習した Generator に入力する潜在変数を画像から推定．次に，Generator はエンコーダの入力と同じになるように画像を生成する
Fig. 4 Overview of general type encoder GAN Inversion pipeline. First, the encoder estimates latent codes to be input to the pre-trained Generator from images. Then, Generator creates the same images as the encoder's input.

2.2 StyleGAN

StyleGAN とは高解像度な画像の生成が可能な生成モデルである．MLP ベースの Mapping Network において確率的に生成される変数 z から変換される潜在変数 w は，画像のスタイルに影響を与える．たとえば顔画像の場合，潜在変数にベクトルを加減算することで，顔の向きや年齢を変化させることが可能である．しかしながら，潜在変数だけではコンテントとスタイルを分離した編集は困難である．また，StyleGAN を改良した StyleGAN2 [16] というモデルがある．このモデルでは StyleGAN で採用されていた Adaptive Instance Normalization（AdaIN）[9] を廃止し，Weight demodulation を用いて正規化，畳み込み処理を行うようにしたものである．Weight demodulation は畳み込み層に入力される特徴マップではなく，畳み込み層の重みに対してスケーリングと正規化を行う．本手法ではこのモデルの Generator を利用する．

2.3 GAN Inversion

GAN Inversion とは GAN の Generator が入力画像を再現できるように，潜在変数を推定することである．GAN Inversion には，潜在変数を直接最適化するアプローチ [14] と，画像を潜在変数にエンコードするエンコーダを利用するアプローチがある [12], [19], [20]．前者は再構成品質が高く，最適化に時間がかかる傾向があり，後者は推定時間が早い反面，再構成品質が低くなる傾向がある．エンコーダは，**図 4** に示すように，Generator が入力と同じ画像を生成するような潜在変数を推定する．Generator は事前に

学習されたモデルであることが多いため，多くのアプローチではエンコーダのみを学習する．

特にエンコーダベースの GAN Inversion アプローチでは，HyperStyle [10] のように，ニューラルネットワークのパラメータを学習するモデルである HyperNetworks [13] で Generator のパラメータを更新することで再構成品質を改善するアプローチも存在する．HyperNetworks は元の入力画像と調整前の Generator で出力された再構成画像を入力に渡し，再構成画像が入力画像を再現するように Generator を調整するパラメータ，残差パラメータを出力する．この残差パラメータで Generator の重みを調整することで再構成品質を向上させている．HyperStyle では，Generator のパラメータ θ を以下の式 (1) のように修正することで，パラメータ $\hat{\theta}$ を与える．

$$\hat{\theta}_l^{i,j} = \theta_l^{i,j}(1 + \Delta_l^{i,j}) \tag{1}$$

$\theta_l^{i,j}$ は，l 番目の Generator に対する畳み込み層における i 番目のフィルタの j 番目のチャネルの重みである．Δ は残差パラメータである．HyperStyle に似た別のアプローチに HyperInverter [11] があるが，本研究では HyperNetworks ベースのアプローチの 1 つである HyperStyle を使用した．

3. 提案手法

本研究の目的は，再構成品質を落とさずにコンテントとスタイルを分離し，より柔軟な編集を可能にすることである．そこで，GAN Inversion という手法に着目した．GAN Inversion では，再構成品質と編集性能の関係はトレードオフであるといわれている [20]．この問題を解決するために，多くのアプローチが考案されている．そのアプローチの 1 つが，トレードオフを解消することを目的とした HyperStyle である．まず HyperStyle の分析から始め，その次にアーキテクチャの詳細と損失関数について説明する．

3.1 HyperStyle の分析

エンコーダベースのアプローチでは，入力画像を潜在変数にエンコードする際に，入力画像の情報を失ってしまう．背景画像の場合，顔画像などの前景画像に比べて多様なデータセットであるため，$\mathcal{W}$ Encoder や Style Transformer のようなエンコーダのみの GAN Inversion では画像の再構成が困難である．$\mathcal{W}$ Encoder は pSp のアブレーション研究で定義された $\mathcal{W}$ 空間（$\mathbb{R}^{512}$）の点である潜在変数を出力するモデルである．$\mathcal{W}$ 空間は $\mathcal{W}$+ 空間（$\mathbb{R}^{n\times512}$）よりも再構成品質が低いが編集性能が高いとされている．この n は Generator の層の数である．HyperStyle では，再構成品質と編集性能の両立を図るために $\mathcal{W}$ 空間を潜在空間として採用している [10]．そのため，HyperStyle では $\mathcal{W}$ 空間の点である潜在変数を出力する $\mathcal{W}$ Encoder を使用している．さらに HypterStyle は HyperNetworks の残差パラメータで Generator の重みを調整することで再構成品質を向上させている．

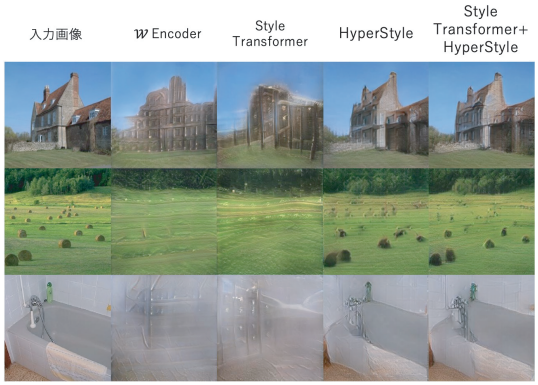


図 5 GAN Inversion の再構成画像例．$\mathcal{W}$ Encoder と Style Transformer の結果はどちらも出力画像がぼやけている（2，3 列目）．HyperStyle を適用したことにより再構成品質が向上し，物体の形状が鮮明になっている（4，5 列目）

Fig. 5 Example of reconstruction quality of GAN Inversion. W Encoder and Style Transformer are the encoder network in GAN Inversion. In both cases, the output image is blurred in the second and third columns, but HyperStyle improves the reconstruction quality and clarifies the shape of objects in the fourth and fifth columns.

HyperStyle を使用する場合とそうでない場合とで再構成品質がどれほど向上しているかを比較するために，3 種類の入力画像に対して各手法の再構成結果を示したのが**図 5** である．Style Transformer [12] の出力は本来，各層に入力する潜在変数がそれぞれ異なる $\mathcal{W}$+空間の点である．本研究では編集品質を重視するため，Style Transformer を使用する際，出力を $\mathcal{W}$ 空間の点になるように調整したものを使用する．$\mathcal{W}$ Encoder と Style Transformer について，図 5 の 2，3 列目に示すように，再構成された画像の結果は入力画像と大きく異なり画像全体がぼやけてしまっている．画像全体のスタイルは残っているが，画像に含まれるコンテントの情報は失われている．入力画像のコンテント情報は潜在変数にエンコードする際に失われており，HyperStyle はそれを回復している．

HyperStyle において，ソース画像から得た残差パラメータを，参照画像から得た残差パラメータに置き換えると，ソース画像のコンテントは参照画像のコンテントになる．たとえば，**図 6** の (c) は，(a) の GAN Inversion において，Generator の残差パラメータを (b) から得られたものに置き換えたものである．コンテントは (b) のものであり，スタイルは (a) と同じである．このように，残差パラメータはコンテントの編集に重要な役割を果たすと考えられる．

3.2 画像編集モデル

3.2.1 モデルアーキテクチャ

本手法の画像編集モデルの概要を**図 7** に示す．本手法は

GAN Inversion のエンコーダベースの手法で，エンコーダから出力される潜在変数から生成した再構成画像と元データの画像を HyperStyle の HyperNetwork に入力し，それらの画像に対する残差パラメータ $\Delta\theta$ を出力する．それを Generator の各畳み込み層のパラメータに足し合わせることで Generator の性能を向上させる．エンコーダに入力した画像を再現できるように Generator は調整される．このとき，エンコーダは事前に学習させたものとなっており，HyperStyle と同時に学習しない．編集品質を向上させるために，エンコーダは HyperStyle でも使用していた $\mathcal{W}$ Encoder を採用する．

3.1 節では残差パラメータが画像のコンテントの編集に寄与すると仮説を立てた．HyperStyle は本来エンコーダの再構成画像と実画像を入力するが，コンテント編集性能を従来の手法から向上させるために，参照画像として実画像の代わりにセマンティックセグメンテーションから合成画像に変換した画像を HyperStyle の入力に用いる．このとき，エンコーダに入力する画像も同様の合成画像である．HyperStyle の入力にセマンティックセグメンテーションから変換された画像を用いることによって画像のコンテントを変えることが可能となる．セマンティックセグメンテーションから合成画像に変換するモデルは，背景画像のデータセットで事前学習した SPADE [7] を使用した．残差パラメータで画像のコンテントが変わる一方で，エンコーダでエンコードした潜在変数 w を調整することで画像のスタイルが変化する．本手法ではコンテント編集をする場合は SPADE の入力のセマンティックセグメンテーション画像を，スタイル編集をする場合は GAN Inversion のエンコーダ（$\mathcal{W}$ Encoder）の入力画像を参照画像に変えることで編集を行う．また，コンテント編集において実画像を参照画像にする場合は SPADE を介さず，HyperStyle に直接入力する．スタイル編集はテキストプロンプトでも行える．その方法は 3.3 節で説明する．

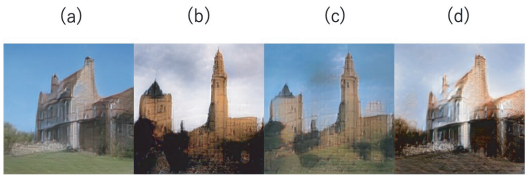


図 6 HyperStyle から取得した残差パラメータによるコンテント情報の確認．入力画像は (a) と (b)．(c) はコンテント画像 (b) の残差パラメータで調整した StyleGAN2 の Generator でスタイル画像 (a) の GAN Inversion した結果．(d) はコンテント画像 (a) の残差パラメータで調整した StyleGAN2 の Generator でスタイル画像 (b) の GAN Inversion した結果

Fig. 6 Confirmation of content information by residue parameters obtained from HyperStyle. Input images are (a) and (b). GAN Inversion (c) of style image (a) using StyleGAN2 Generator adjusted by the residual parameters according to content image (b). GAN Inversion (d) of style image (b) in the Generator of StyleGAN2 adjusted by the residual parameters according to content image (a).

SPADE によって変換された画像は実画像のデータセットに比べてスタイルの多様性が劣っているという側面がある．そこで画像編集モデルを学習する際，データセットはセマンティックセグメンテーションから変換された画像だけでなく従来の実画像データセットも加える．実画像で学習するときは，SPADE を用いず，HyperStyle に直接実画像を入力する．

また HyperStyle で調整する StyleGAN2 のパラメータは高次元ピクセルデータから RGB データに変換する tRGB 機構 [16] を考慮していなかった．本研究では従来の畳み込み層のパラメータだけでなく，Generator の 4，5 層目の tRGB 機構のパラメータも考慮して学習する．実際に画像編集をするときは tRGB 機構に残差パラメータを足し合わせないようにして編集を行う．

3.2.2 損失関数

エンコーダのモデルは事前に学習させており，本モデルでは HyperStyle のネットワークを学習する．このモデル

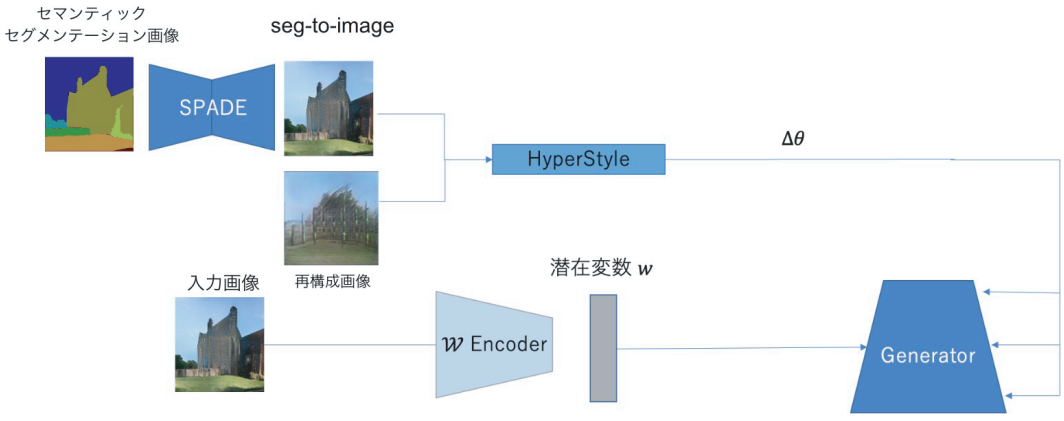


図 7 提案手法のモデルアーキテクチャ図

Fig. 7 Our model architecture.

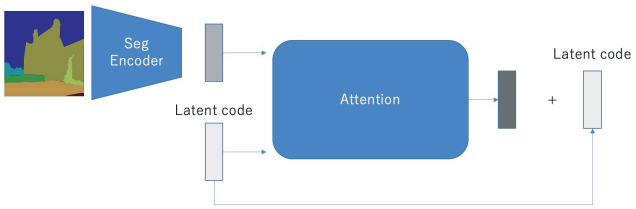


図 8 テキストによるスタイル編集モデル
Fig. 8 Text-based style editing model.

の損失関数は，従来の HyperStyle で使用されていたものと同様である．その損失関数は以下の式のようになる．

$$L = \lambda_2 L_2(x, \hat{y}) + \lambda_{sim} L_{sim}(x, y, \hat{y}) + \lambda_{perc} L_{LPIPS}(x, \hat{y}) \tag{2}$$

x，y は本タスクにおいては同一のものでありどちらも元のデータセットの画像である．$\hat{y}$ は，パラメータを調整した StyleGAN の出力である．λ_2，λ_{sim}，λ_{perc} はそれぞれ L_2，L_{sim}，L_{LPIPS} の重みパラメータである．L_2 は L2 距離，L_{LPIPS} は VGG [30] などの画像分類モデルの特徴量を用いた人間の視覚に基づく知覚損失である．L_{sim} では顔画像を生成・編集するタスクの場合，アイデンティティベースの顔認識モデルを利用した類似損失を使用するケースが多い．しかし本研究では背景画像の編集を中心に行うため，MoCo ベースの類似損失を用いる [19], [20].

3.3 テキストによるスタイル編集モデル

3.3.1 モデルアーキテクチャ

テキストによる画像のスタイル編集モデルのアーキテクチャを図 8 に示す．Seg Encoder はセマンティックセグメンテーションから潜在変数と同次元の特徴量にエンコードするモデルである．テキストによる画像のスタイル編集モデルでは，セマンティックセグメンテーション，Attention 機構 [29] を使用している．セマンティックセグメンテーションを使用すれば，テキストにおいても画像のコンテントを考慮した編集ができるのではないかと考えた．セマンティックセグメンテーションをエンコードした特徴量と編集対象の画像をエンコードした潜在変数を Attention 機構に入力する．Attention 機構は潜在変数と同次元のベクトルを出力し，StyleCLIP のように，その Attention 機構の出力を元の潜在変数と足し合わせる．そして，テキストの内容に即した潜在変数になるように Attention 機構と Seg Encoder を学習する．元の潜在変数 w に対して，編集後の潜在変数 w' は次の式 (3) で定式化される．

$$w' = w + \lambda * Attention(E_s(s), w) \tag{3}$$

E_s は Seg Encoder，s はセマンティックセグメンテーション画像であり，λ は Attention 機構の出力に対する重みパラメータである．$Attention$ は図 8 の機構である．Attention の query は Seg Encoder の出力，key と value は元の潜在変数 w とした．Attention 機構を導入した意図は，セマンティックセグメンテーションの特徴量から潜在変数の特定の部分を編集してコンテントを考慮した編集を可能にするためである．

3.3.2 損失関数

データにはない背景の画像を生成するうえで Discriminator を使用しないため通常の GAN で用いられる adversarial loss を使用しない．全体のモデルの損失関数 L は以下の式 (4) で表す．そして，この L を最小化するように Attention 機構と Seg Encoder を学習する．

$$L = \lambda_{clip} L_{clip}(t, \hat{y}) + \lambda_{sim} L_{sim}(x, y, \hat{y}) + \lambda_2 ||w' - w||_2 \tag{4}$$

λ_{clip}，λ_{sim}，λ_2 はそれぞれ L_{clip}，L_{sim}，w'，w 間の L2 距離に対する重みパラメータである．1 から CLIP の出力を引いたものが L_{clip} となる．L_{sim} は 3.2.2 項のものと同様に背景の画像を用いるため MoCo ベースの類似損失を用いる．

4. 画像生成・編集実験

4.1 実験設定

本実験では多くの背景画像を含むデータセットでモデルを学習させる必要がある．さらにセマンティックセグメンテーション画像のデータも必要である．SUN database [32] は，コンピュータビジョンにおいてシーン分類を行うためのベンチマークのデータセットである．シーンとは，「人間がその中で行動できる場所，または，人間が移動できる場所」を指す．SUN database はあらゆるシーンの画像が含まれるように構築されており，約 900 のシーンを定義している．ADE20K データセットは，この 900 シーンすべてを網羅する画像が収集されている [18]．背景とシーンが必ずしも一致するわけではないが，ADE20K には様々なシーンの画像が含まれていることから，多くの種類の背景画像が含まれていると考えられる．実際，ADE20K は，明確な形状を持つオブジェクト（車や人など）や，無定形の背景画像を持つもの（草や空など）など，すべての視覚的概念を網羅する目的で構築されている．そのため，本研究では ADE20K を対象として実験を行った．このデータセットは，背景画像の学習データ 20,210 枚と検証データ 2,000 枚で構成されている．このデータセットで StyleGAN2 [16] Generator と $\mathcal{W}$ Encoder をそれぞれ，200,000 イテレーション，250,000 イテレーションで事前学習したモデルを使用する．出力画像の解像度は 256 × 256 の画像であり，これは入力画像とセマンティックセグメンテーション画像の両方についても同様である．本手法は Pytorch で実装されている．最適化手法として Ranger [21] を採用している．

提案した画像編集モデルと本実験で使用した既存手法のモデルの損失関数は $\lambda_2 = 1.0$，$\lambda_{perc} = 0.8$，$\lambda_{sim} = 0.1$ と設定した．最適化手法の学習率，そしてイテレーション数については**表 1** に示す．また，本実験では編集品質を重視するために Style Transformer については出力を $\mathcal{W}$ 空間の点になるように調整したものを使用する．

テキストによるスタイル編集モデルについて，損失関数の重みは $\lambda_{CLIP} = 1.0$，$\lambda_{sim} = 0.1$，$\lambda_2 = 0.05$，最適化手法の学習率は 0.1 と設定し，50,000 イテレーションで学習した．

4.2 再構成品質

4.2.1 定性評価

生成画像の再構成品質を定性的に確かめる．実画像における再構成画像の結果を**図 9** に示す．これは $\mathcal{W}$ Encoder，pSp，Style Transformer の 3 つのエンコーダのみの方式およびベースラインとなる HyperStyle と提案手法を比較したものである．その結果，提案手法は StyleTrasnformer などのエンコーダのみの方式の手法よりも入力画像をより再現しており，ベースラインである HyperStyle とはほとんど同等の再構成品質になっている．また，セマンティックセグメンテーションから合成した画像（seg-to-image）の場合についての再構成品質の比較結果は付録 A.1 に示す．

4.2.2 定量評価

本実験では，再構成画像の品質を L2 距離，LPIPS [27]，PSNR，MS-SSIM [23] で定量的に評価した．まず，既存手法との比較を行う．提案手法に対して，本手法で用いた $\mathcal{W}$ Encoder，pSp，Style Transformer の 3 つのエンコーダのみの方式，ベースラインとなる HyperStyle と比較した．その結果を**表 2** に示す．

その結果，HyperStyle に比べ，再構成品質は若干劣る．しかしながら，3 つのエンコーダのみの方式と比較すると再構成品質は定量的に優位な結果となっている．特に，再構成品質が高いとされる $\mathcal{W}+$ 空間で GAN Inversion を行う pSp に対しても提案手法の方が再構成品質が高い．

次に提案手法で用いるエンコーダの比較を行う．エンコーダは $\mathcal{W}$ Encoder と Style Transformer の 2 つで比較した．その結果を**表 3** に示す．提案手法で採用した $\mathcal{W}$ Encoder は Style Trasnformer よりも若干，再構成品質が劣る結果となっている．

最後に，Generator の tRGB 機構における残差パラメータの学習について比較を行う．提案手法では，4，5 層の tRGB 機構のパラメータを考慮して学習しているのに対して，すべての層の tRGB 機構を考慮して学習した場合について再構成品質の比較を行った．その結果を**表 4** に示す．この場合でも tRGB 機構に残差パラメータを足し合わせないようにしている．

その結果，4，5 層目の tRGB 機構のパラメータも考慮した場合，つまり提案した手法の方がすべての tRGB 機構のパラメータも考慮した場合と比較して再構成品質が定量

表 1 画像編集モデル（提案手法）と既存手法のモデルでの学習率，イテレーション数の設定

Table 1 Learning rate and number of iterations in the proposed image editing model and in the model of the existing method.

method	学習率	イテレーション
提案手法	0.0004	200,000
HyperStyle（ベースライン）	0.0004	200,000
pSp	0.0001	250,000
Style Transformer	0.0001	200,000
$\mathcal{W}$ Encoder	0.0001	250,000

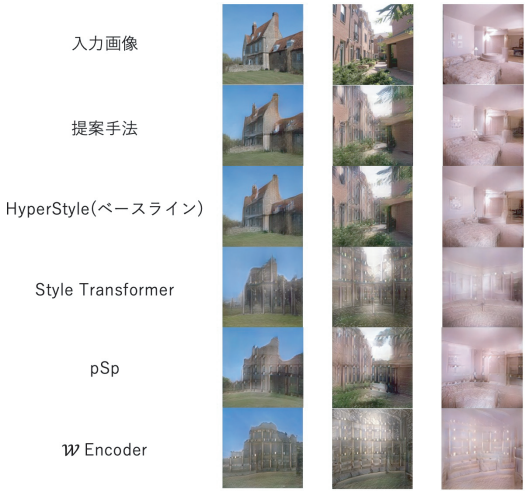


図 9 実画像における再構成画像の比較結果

Fig. 9 Comparison results of reconstructed images in real images.

表 2 既存手法との再構成品質の比較

Table 2 Comparison of reconstruction quality with existing methods.

	再構成品質			
method	L2(↓)	LPIPS(↓)	PSNR(↑)	MS-SSIM(↑)
提案手法	0.05429	0.24280	19.37055	0.64831
HyperStyle（ベースライン）	**0.04389**	**0.19549**	**20.13440**	**0.67653**
Style Transformer	0.08499	0.42089	17.09287	0.40741
pSp	0.06722	0.31486	18.18414	0.52306
$\mathcal{W}$ Encoder	0.11540	0.48089	15.77095	0.28603

表 3 提案手法におけるエンコーダ間の再構成品質の比較
Table 3 Comparison of reconstruction quality between encoders in the proposed method.

	再構成品質			
method	L2(↓)	LPIPS(↓)	PSNR(↑)	MS-SSIM(↑)
$\mathcal{W}$ Encoder の場合	0.05429	0.24280	19.37055	0.64831
Style Transformer の場合	0.04883	0.21527	19.59198	0.66018

表 4 tRGB の残差パラメータに関する比較（テスト時では tRGB の残差パラメータを足し合わせない）
Table 4 Comparison on tRGB residual parameters. For testing, no addition of tRGB residual parameter.

	再構成品質			
method	L2(↓)	LPIPS(↓)	PSNR(↑)	MS-SSIM(↑)
4，5 層を考慮して学習（**提案手法**）	0.05429	0.24280	19.37055	0.64831
すべての層を考慮して学習	0.07596	0.25653	17.54552	0.59844

的に高いことが分かる．

4.3 画像編集品質

画像編集の実験として次の 2 つを行う．

- 元画像のスタイルを維持したままコンテントを編集（コンテント編集）
- 元画像のコンテントを保持したままスタイルのみを編集（スタイル編集）

コンテント編集では編集対象画像のスタイルが残っており，コンテントだけが変化しているかどうかを確かめる．それに対して，スタイル編集では編集対象画像のコンテントが残っており，スタイルだけが変化しているかどうかを確かめる．コンテント編集，スタイル編集どちらも，実画像の場合と seg-to-image の場合の両方で実験した．これら 2 つの場合の実験結果についてはそれぞれ異なる編集の結果として考える．またコンテント編集については GAN Inversion のエンコーダの入力を実画像，HyperStyle ネットワークの入力を seg-to-image にした場合についても検証した．編集性能に関しては，定量的に評価することが困難であることが先行研究において示されている [31]．既存の編集性能の評価は顔データと顔属性に対する編集性能の評価に焦点を当てるなど，顔の同一性の保持を測定しており，顔以外のすべての画像領域には適用できない可能性がある [31]．そのため背景画像を対象にする本研究においては定性的な評価のみを行った．コンテント，スタイル編集ではソース画像 50 枚，参照画像 50 枚による編集結果を確認したが，紙面の都合上本論文では 3 枚の編集結果のみを示す．テキストによるスタイル編集では 500 枚の編集結果を確認したが，同様に 3 枚の編集結果のみを示す．こちらの編集に用いたプロンプトは “desert”，“ocean”，“red sky” の 3 種類である．これらのプロンプトは，1 章で説明した絵柄や画風にあたる．これら 50 枚および 500 枚の編集結

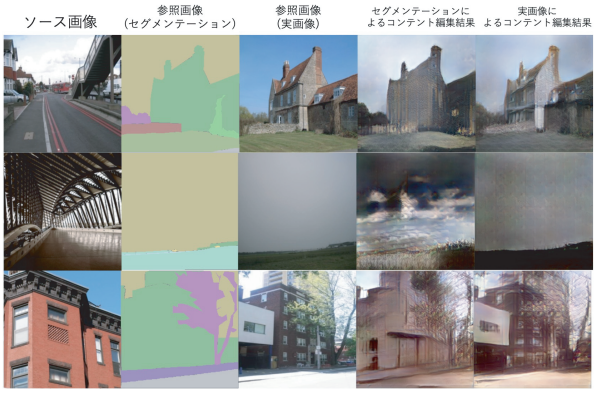


図 10 提案手法による，1 列目のソース画像に対するコンテント編集の結果．コンテント編集の参照画像として，セマンティックセグメンテーションと実画像を入力（2，3 列目）．4，5 列目がそれぞれの入力に対するコンテント編集結果
Fig. 10 Result of content editing for the source image in the first row by the proposed method. The second and third rows are the input semantic segmentation and real images as the reference images for content editing. Columns 4 and 5 show the result of content editing for each input.

果を確認したところ，コンテント編集ではソース画像のスタイルが維持されないまたはコンテントが参照画像のコンテントに編集されていない，スタイル編集ではソース画像のコンテントが維持されていないまたはスタイルが参照画像のスタイルに編集されていないというものはなかった．

4.3.1 コンテント編集

まず実画像に対して，コンテント編集の参照画像をセマンティックセグメンテーション，つまり seg-to-image の場合と実画像にした場合について，それぞれのコンテント編集の結果を確かめた．その結果を**図 10** に示す．どちらの場合においても入力どおりのコンテントに編集されている．

次に，提案手法における実画像間のコンテントのミキシング結果を**図 11** に示す．これらの図の 1 列目と 1 行目

図 11 提案手法における，実画像間のコンテントミキシングの結果．1 列目と 1 行目はそれぞれソース画像，参照画像であり，各列の画像のコンテントは，参照画像のコンテントに編集される

Fig. 11 Result of content mixing between real images in the proposed method. The first column and the first row are the source and reference images, respectively, and the content of the image in each row is edited to the content of the reference image.

はそれぞれソース画像，参照画像であり，各列の画像のコンテントは，参照画像のコンテントに編集される．また，seg-to-image 間におけるコンテントミキシングの結果は付録 A.2 に示す．

さらに，提案手法に対してベースラインである HyperStyle と提案手法のエンコーダを Style Transformer に変更したものでコンテントの編集の比較を行った．その結果を以下の**図 12** に示す．実画像における結果を見ると，ベースライン手法では，参照画像のスタイルまで少々反映されてしまっており，特に 1 行目ではソース画像のスタイルの面影があまり残っていない．それに対して提案手法では，参照画像のスタイルはほとんど反映されず，コンテントのみの編集ができている．

4.3.2 スタイル編集

スタイル編集についてまず，提案手法における実画像間のスタイルのミキシング結果を**図 13** に示す．これらの図の 1 列目と 1 行目はそれぞれソース画像，参照画像である．各列の画像のスタイルは，参照画像のスタイルに編集される．また，seg-to-image 間におけるスタイルミキシングの結果は付録 A.2 に示す．

さらに提案手法に対してベースラインである HyperStyle と提案手法のエンコーダを Style Transformer に変更したものでスタイル編集の比較を行った．その比較結果を**図 14** に示す．この結果を見ると，ベースライン手法では，参照画像のスタイルがあまり反映されておらず，特に 1 行目ではそれが顕著である．それに対して提案手法では，参照画像のコンテントを反映することなく，スタイルのみの編集ができている．

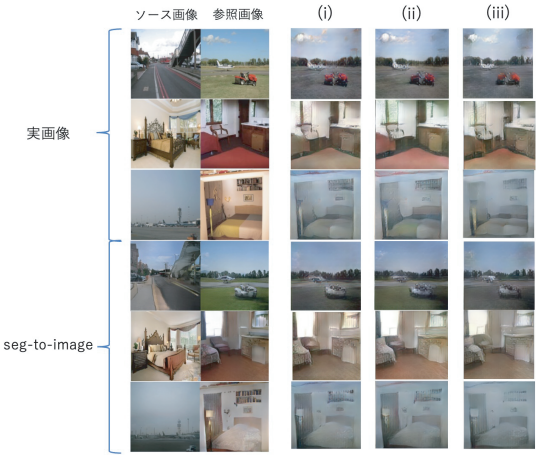


図 12 実画像，seg-to-image におけるコンテント編集の比較結果．(i) が提案手法，(ii) がベースラインの HyperStyle，(iii) が提案手法に対してエンコーダを Style Transformer に変更したときの 1 列目のソース画像に対するコンテント編集結果

Fig. 12 Comparative results of content editing on real images and seg-to-images. For the first row of source images, (i) is the content editing result of the proposed method, (ii) is the content editing result of the baseline approach, HyperStyle, and (iii) is the content editing result of the proposed method in which the encoder is changed to Style Transformer.

図 13 提案手法における，実画像間のスタイルミキシングの結果．1 列目と 1 行目はそれぞれソース画像，参照画像であり，各列の画像のスタイルは，参照画像のスタイルに編集される

Fig. 13 Result of style mixing between real images in the proposed method. The first column and the first row are the source and reference images, respectively, and the style of the image in each row is edited to the style of the reference image.

4.3.3 テキストによるスタイル編集

本実験では，提案したテキストによるスタイル編集モデルで，テキストで画像のスタイルが編集可能かどうかを確認する．その結果を**図 15** に示す．1 列目が編集する前の再構成画像であり，2 列目以降は “desert”，“ocean”，“red sky” の 3 つのテキストプロンプトによるスタイル編集結果である．その結果，画像のコンテントを保ちつつ，

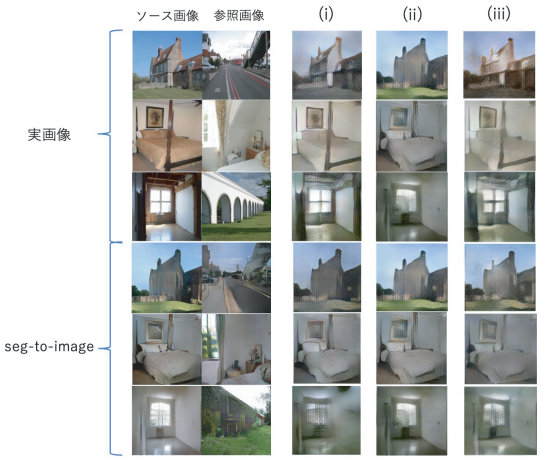


図 14 実画像，seg-to-image におけるスタイル編集の比較結果．(i) が提案手法，(ii) がベースラインの HyperStyle，(iii) が提案手法に対してエンコーダを Style Transformer に変更したときの 1 列目のソース画像に対するスタイル編集結果

Fig. 14 Comparative results of style editing on real images and seg-to-images. For the first row of source images, (i) is the style editing result of the proposed method, (ii) is the style editing result of the baseline approach, HyperStyle, and (iii) is the content style result of the proposed method in which the encoder is changed to Style Transformer.

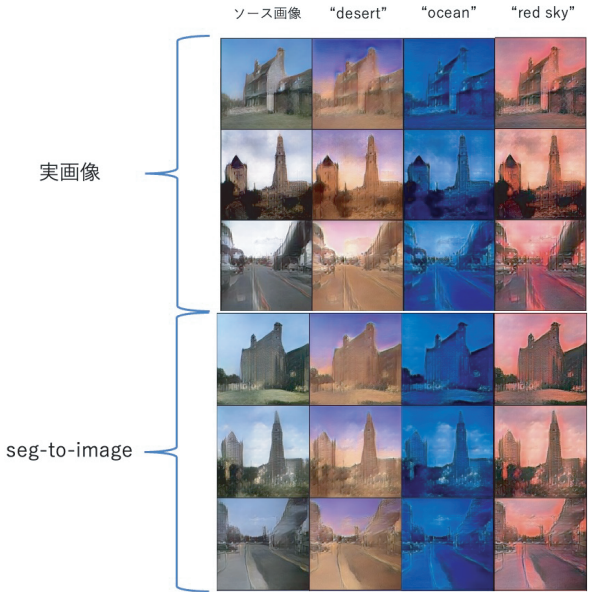


図 15 提案手法におけるテキストによる実画像，seg-to-image のスタイル編集の結果

Fig. 15 Results of editing the style of real images with text in the proposed method.

スタイルがテキストの内容に即したものとなっている．また，比較対象を提案した画像編集モデルにおいて，テキストによるスタイル編集モデルを StyleCLIP に変えたものとしている．その編集結果を**図 16** に示す．テキストの内容が “desert” のとき（2 列目），地面は砂漠のように黄色かつ空の色は維持すべきである．提案モデルと既存手法の

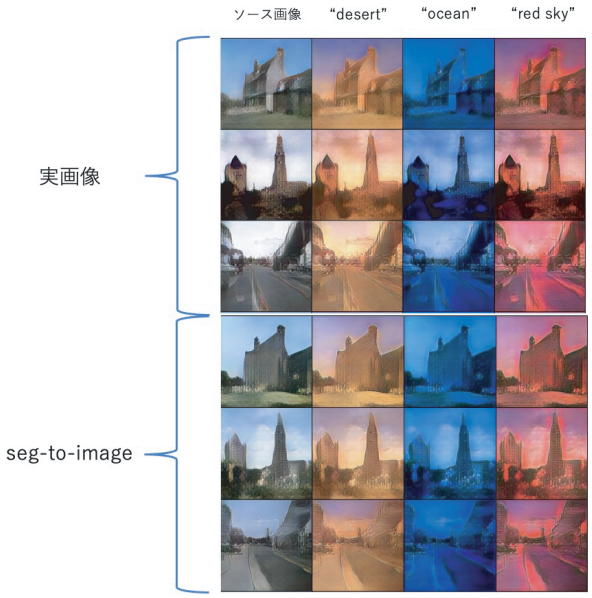


図 16 StyleCLIP におけるテキストによる実画像，seg-to-image のスタイル編集の結果

Fig. 16 Results of editing the style of real images with text in StyleCLIP.

StyleCLIP の結果を比較すると，提案手法の方が空が青くなっている．またテキストの内容が “red sky” のとき（4 列目），実画像と seg-to-image 両方の場合で StyleCLIP では空の青みが残ってしまっているが，提案手法にはそれがなくテキストの内容に忠実である．

5. 考察

再構成品質について考察する．実験結果から Style Transformer や pSp といった既存のエンコーダのみの手法と比べると定量的，定性的ともに再構成品質が高いことが分かる．しかしながら，再構成品質における定量評価から，提案手法の再構成品質はベースラインの手法と比べてやや劣っている結果となった．次にコンテント編集とスタイル編集について考察する．まず，コンテント編集は，1 章で定義したとおり，画像のスタイルを維持したままコンテントのみを編集するということである．実験結果を見るとベースラインの HyperStyle はスタイルも参照画像に近いスタイルに変わっていたが，提案手法ではスタイルはソース画像のままとなっており，スタイルを維持したままコンテントのみが編集がされている．したがって，コンテント編集が可能であると考える．次にスタイル編集は，1 章で定義したとおり，画像のコンテントを維持したままスタイルのみを編集することである．提案手法ではコンテントを維持したままスタイルのみが編集されている．テキストによる編集でも，コンテントが変わらずスタイルのみが編集されている．このことから，スタイル編集が可能だと考える．このことによってベースラインである HyperStyle と比べて，提案手法ではより高い編集性能を実現できたと考える．こ

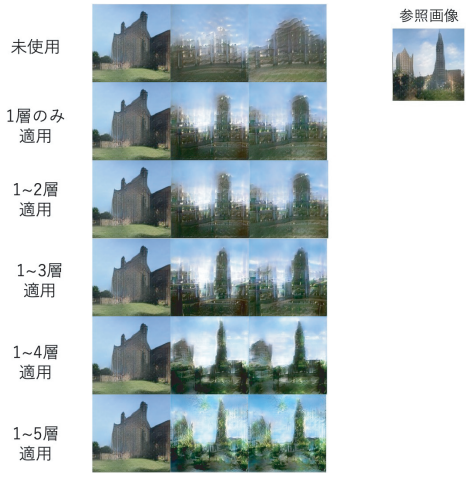


図 17 HyperStyle における残差パラメータの確認．1 列目の画像は編集対象の画像，2 列目はコンテント編集における参照画像，3 列目は 1 列目の画像に対してコンテント編集した結果．1 行目では残差パラメータを使用せず，2 行目以降では低い層からだんだんと残差パラメータを加えた結果である．この図では 5 層目までを記している

Fig. 17 Checking the residual parameters in HyperStyle. The first column shows the image to be edited, the second column shows the reference image for content editing, and the third column is the result of content editing on the image in the first column. The first row show the result of content editing on the image without the residual parameter. The second and subsequent rows show the content editing result of adding the residual parameter to Generator gradually starting from the lowest layer. In the figure, up to the fifth layer is shown.

れらから，提案手法は柔軟性のある編集はできたが，再構成品質ではやや劣る結果になった．しかしながら，背景画像では前景画像に比べてカメラと被写体の間の絶対的な距離が遠く，画像の鮮明さは前景画像よりも重視されないと考えている．したがって，既存手法よりも柔軟な編集が可能である本手法にはメリットがあると考えられる．

テキストによるスタイル編集について，提案したモデルは “red sky” において，空の色が StyleCLIP よりもテキストの内容に忠実だった．これは提案したモデルの入力にセマンティックセグメンテーションを渡しており，その特徴量が影響したものだと考える．

また，seg-to-image 画像において，似たようなスタイルの画像に変換される場合がある．それはセマンティックセグメンテーションから実画像に合成するモデルとして使用した SPADE の性能によるものと考えられる．SPADE よりもセマンティックセグメンテーションから多様なスタイルの画像を生成できる手法を用いることが必要だと考える．それらのデータで学習すれば，seg-to-image においても多様なスタイルの編集ができ，より幅広いコンテントとスタイルの編集が可能になると考える．

提案手法に対してエンコーダを $\mathcal{W}$ Encoder にしたものと Style Transformer にしたもの（図 12，14 の 3，5 列目）を比較すると，スタイル編集において図 14 の 1 行目では Style Transformer をエンコーダとした手法で参照画像のスタイルが反映されず，ソース画像のスタイルがやや残っており，さらにモヤのようなものが出てしまっている．また図 12 のように seg-to-image におけるコンテント編集結果でも同様にモヤのようなものが出ている．このような結果から，本手法において $\mathcal{W}$ Encoder をエンコーダとして使用した場合の方が編集性能が高いと考える．

ベースラインではコンテントとスタイルを分離した編集がうまくできなかった．そこでベースラインにおける，Generator の各層における残差パラメータの影響について調査をした．残差パラメータを Generator の層ごとに段階的に追加し，スタイルが影響してしまう場所を確認する．その結果が**図 17** のようになった．この図から，4，5 層目の残差パラメータがスタイル編集がうまく反映されない大きな要因になっていると考えられる．提案手法では 4，5 層では学習時に tRGB 機構を考慮して学習した．tRGB が色に関する畳み込みであるため，この機構がコンテントとスタイルの分離に影響したと考える．

6. まとめ・今後の展望

本研究では，背景画像の多様性に起因する GAN Inversion の再構成品質低下の問題を，HyperNetworks を用いた Generator のパラメータ更新手法である HyperStyle により解決する．また，テキストでは困難であったコンテントの編集が，HyperNetworks の残差パラメータを用いることで実現可能であることが確認できた．本手法は，コンテントとスタイルを分離した柔軟な背景画像編集を可能にする一方で，再構成画像の品質は既存手法である HyperStyle とはやや劣ることが確認された．

今後の展望として，第 1 にスタイル編集性能の向上である．seg-to-image 画像では別のセマンティックセグメンテーション画像を入力したとしても似たようなスタイルの画像が生成される場合があり，セマンティックセグメンテーションによる画像編集において，スタイル編集の多様性が不足してしまう．本手法ではセマンティックセグメンテーションから画像に合成する手法として SPADE を使用したが，多様なスタイルに対応できるようにする必要がある．そのため新たに多様なスタイルに対応できるようなセマンティックセグメンテーションから画像に合成する手法を考案していく．

第 2 に，画像の生成品質の向上である．近年，Diffusion Model [28] による画像生成の技術の発展が著しい．そのモデルは生成画像の品質が高く，多様な生成が可能である．本手法に Diffusion Model を導入して生成画像の品質向上を検討する．

さらに本研究では ADE20K データセットのみで実験を行ったが，他のデータセットについても評価を行うことが望ましいと考えられるため，将来課題として，COCO [*2] など他のデータセットでの評価を行うこととする．

謝辞 本研究は JSPS 科研費 JP21H03496，JP22K12157 の助成を受けたものです．

参考文献

[1] Ishihata, S., Orihara, R., Sei, Y., Tahara, Y. and Ohsuga, A.: Background image editing with hyperstyle and semantic segmentation, *International Conference on Agents and Artificial Intelligence* (*ICAART*) (2023).

[2] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y.: Generative adversarial nets, *Neural Information Processing Systems* (*NIPS*) (2014).

[3] Radford, A., Metz, L. and Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks, *International Conference on Learning Representations* (*ICLR*) (2015).

[4] Isola, P., Zhu, J.-Y., Zhou, T. and Efros, A.A.: Image-to-image translation with conditional adversarial networks, *Computer Vision and Pattern Recognition* (*CVPR*) (2017).

[5] Ronneberger, O., Fischer, P. and Brox, T.: Unet: Convolutional networks for biomedical image segmentation, *International Conference on Medical Image Computing and Computer-assisted Intervention*, pp.234–241 (2015).

[6] Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J. and Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans, *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (2018).

[7] Park, T., Liu, M.-Y., Wang, T.-C. and Zhu, J.-Y.: Semantic image synthesis with spatially-adaptive normalization, *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), pp.2337–2346 (2019).

[8] Ioffe, S. and Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift, *International Conference on Machine Learning* (*ICML*) (2015).

[9] Huang, X. and Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization, *ICCV* (2017).

[10] Alaluf, Y., Tov, O., Mokady, R., Gal, R. and Bermano, A.: Hyperstyle: Stylegan inversion with hypernetworks for real image editing, *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), pp.18490–18500, IEEE Computer Society (2022).

[11] Dinh, T.M., Tran, A.T., Nguyen, R. and Hua, B.-S.: Hyperinverter: Improving stylegan inversion via hypernetwork, *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*) (2022).

[12] Hu, X., Huang, Q., Shi, Z., Li, S., Gao, C., Sun, L. and Li, Q.: Style transformer for image inversion and editing, arXiv preprint arXiv:2203.07932 (2022).

[13] Ha, D., Dai, A.M. and Le, Q.V.: Hypernetworks, *International Conference on Learning Representations* (2017).

[14] Abdal, R., Qin, Y. and Wonka, P.: Image2stylegan: How to embed images into the stylegan latent space?, *Proc. IEEE/CVF International Conference on Computer Vision* (*ICCV*) (Oct. 2019).

[15] Karras, T., Laine, S. and Aila, T.: A style-based generator architecture for generative adversarial networks, *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*) (June 2019).

[16] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J. and Aila, T.: Analyzing and improving the image quality of StyleGAN, *Proc. CVPR* (2020).

[17] Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D. and Lischinski, D.: Styleclip: Text-driven manipulation of stylegan imagery, *Proc. IEEE/CVF International Conference on Computer Vision* (*ICCV*), pp.2085–2094 (Oct. 2021).

[18] Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A. and Torralba, A.: Scene parsing through ade20k dataset, *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (2017).

[19] Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S. and Cohen-Or, D.: Encoding in style: a stylegan encoder for image-to-image translation, *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*) (June 2021).

[20] Tov, O., Alaluf, Y., Nitzan, Y., Patashnik, O. and Cohen-Or, D.: Designing an encoder for stylegan image manipulation, arXiv preprint arXiv:2102.02766 (2021).

[21] Wright, L.: Ranger – a synergistic optimizer, available from ⟨https://github.com/lessw2020/Ranger-Deep-Learning-Optimizer⟩ (2019).

[22] Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X. and He, X.: Attngan: Fine-grained text to image generation with attentional generative adversarial networks (2018).

[23] Wang, Z., Simoncelli, E.P. and Bovik, A.C.: Multiscale structural similarity for image quality assessment, *The 37 Asilomar Conference on Signals, Systems & Computers*, pp.1398–1402 (2003).

[24] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G. and Sutskever, I.: Learning transferable visual models from natural language supervision, arXiv preprint arXiv:2103.00020 (2021).

[25] Karras, T., Aila, T., Laine, S. and Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation, *International Conference on Learning Representations* (2018).

[26] Zhu, P., Abdal, R., Qin, Y. and Wonka, P.: Sean: Image synthesis with semantic region-adaptive normalization, *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*) (June 2020).

[27] Zhang, R., Isola, P., Efros, A.A., Shechtman, E. and Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric, *CVPR* (2018).

[28] Rombach, R., Blattmann, A., Lorenz, D., Esser, P. and Ommer, B.: High-resolution image synthesis with latent diffusion models (2021).

[29] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I.: Attention is all you need, *Advances in Neural Information Processing Systems* (*NIPS*) (2017).

[30] Simonyan, K. and Zisserman, A.: Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

[31] Xia, W., Zhang, Y., Yang, Y., Xue, J.-H., Zhou, B. and

[*2] https://cocodataset.org/

Yang, M.-H.: Gan inversion: A survey, arXiv preprint arXiv:2101.05278 (2021).
[32] Xiao, J., Hays, J., Ehinger, K.A., Oliva, A. and Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo, *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp.3485–3492, IEEE (2010).

付　　録

A.1 再構成品質

seg-to-image における再構成品質の比較結果を**図 A·1** に示す．この結果でも実画像の場合と同様に入力画像を再現しており，ベースラインである HyperStyle とはほとんど同等の再構成品質になっている．

A.2 編集品質

seg-to-image におけるコンテントミキシングおよびスタイルミキシングの結果を**図 A·2** および **図 A·3** に示す．seg-to-image におけるコンテント編集の結果では実画像の結果と同様に，ベースラインはスタイルも反映されてしまう傾向にあったが，提案手法では，参照画像のスタイルはベースラインよりもあまり反映されずコンテントのみの編集ができている．しかしながら，実画像における編集結果と比べるとスタイルが反映されてしまう傾向にある．

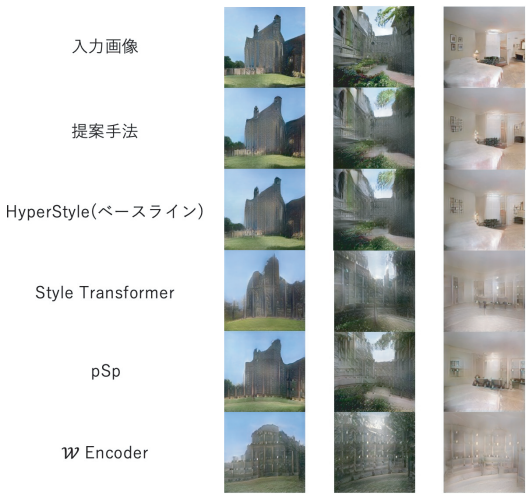


図 A·1 seg-to-image における再構成画像の比較結果
Fig. A·1 Comparison results of reconstructed images in seg-to-images.

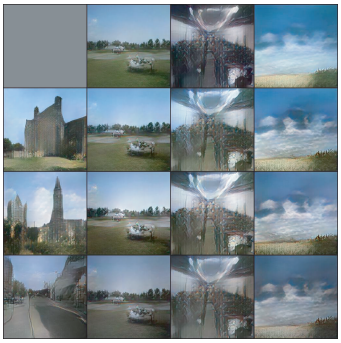


図 A·2 提案手法における，seg-to-image 間のコンテントミキシングの結果．1 列目と 1 行目はそれぞれソース画像，参照画像であり，各列の画像のコンテントは，参照画像のコンテントに編集される
Fig. A·2 Result of content mixing between seg-to-images in the proposed method. The first column and the first row are the source and reference images, respectively, and the content of the image in each row is edited to the content of the reference image.

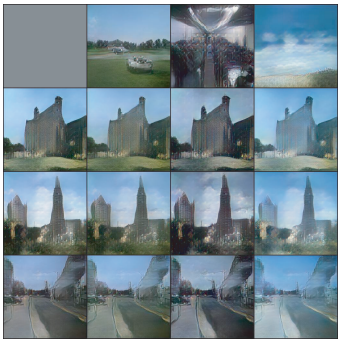
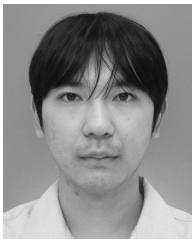


図 A·3 提案手法における，seg-to-image 間のスタイルミキシングの結果．1 列目と 1 行目はそれぞれソース画像，参照画像であり，各列の画像のスタイルは，参照画像のスタイルに編集される
Fig. A·3 Result of style mixing between seg-to-images in the proposed method. The first column and the first row are the source and reference images, respectively, and the style of the image in each row is edited to the style of the reference image.



石幡 柊介

1999 年生．2021 年電気通信大学情報理工学域卒業．同年電気通信大学大学院情報理工学研究科情報学専攻入学．画像生成・編集に関する研究に従事．

折原 良平 （正会員）

1988 年筑波大学大学院工学院研究科電子・情報工学専攻博士前期課程修了．同年（株）東芝入社．2019 年東芝メモリ（株）（現，キオクシア（株））入社．現在，同社メモリ技術研究所デジタルトランスフォーメーション技術研究開発センター技監．1993〜1995 年 University of Toronto, Department of Industrial Engineering 客員研究員．2010 年より電気通信大学情報システム学研究科客員教授．発想支援技術，類推，機械学習，データ・テキストマイニングの研究に従事．2009 年度人工知能学会論文賞，2010 年度人工知能学会功労賞，2012 年度情報処理学会活動賞，2016 年度人工知能学会現場イノベーション賞金賞，2020 年度同賞銀賞受賞．2015 年度情報処理学会フェロー．2017〜2019 年人工知能学会副会長．博士（工学）．

清 雄一 （正会員）

1981 年生．2009 年東京大学大学院情報理工学系研究科博士後期課程修了．同年（株）三菱総合研究所入社．2013 年より電気通信大学．現在，同大学大学院情報理工学研究科教授．博士（情報理工学）．エージェント，プライバシ保護技術等の研究に従事．2016 年度土木学会水工学論文賞，情報処理学会論文賞受賞．電子情報通信学会，日本ソフトウェア科学会，IEEE Computer Society 各会員．

田原 康之

1966 年生．1991 年東京大学大学院理学系研究科数学専攻修士課程修了．同年（株）東芝入社．1993〜1996 年情報処理振興事業協会に出向．1996〜1997 年英国 City 大学客員研究員．1997〜1998 年英国 Imperial College 客員研究員．2003 年国立情報学研究所着任．2008 年より電気通信大学准教授．博士（情報科学）（早稲田大学）．エージェント技術，およびソフトウェア工学等の研究に従事．日本ソフトウェア科学会会員．

大須賀 昭彦 （正会員）

1958 年生．1981 年上智大学理工学部数学科卒業．同年（株）東芝入社．同社研究開発センター，ソフトウェア技術センター等に所属．1985〜1989 年（財）新世代コンピュータ技術開発機構（ICOT）出向．2007 年より電気通信大学．現在，同大学大学院情報理工学研究科教授．2017 年より同大学大学院情報システム学研究科研究科長併任．2012 年より国立情報学研究所客員教授兼任．工学博士（早稲田大学）．ソフトウェア工学，エージェント，人工知能の研究に従事．1986 年度および 2016 年度情報処理学会論文賞，2013 年度人工知能学会研究会優秀賞，2014 年度同学会功労賞，2018 年度電子情報通信学会 ISS 活動功労賞受賞．IEEE Computer Society Japan Chapter Chair，人工知能学会理事，日本ソフトウェア科学会理事，同学会監事等を歴任．電子情報通信学会，人工知能学会，日本ソフトウェア科学会，電気学会，IEEE Computer Society 各会員．本会フェロー．